

C# Object and Class

Since C# is an object-oriented language, program is designed using objects and classes in C#.

C# Object

In C#, Object is a real world entity, for example, chair, car, pen, mobile, laptop etc.

In other words, object is an entity that has state and behavior. Here, state means data and behavior means functionality.

Object is a runtime entity, it is created at runtime.

Object is an instance of a class. All the members of the class can be accessed through object.

Let's see an example to create object using new keyword.

```
Student s1 = new Student();//creating an object of Student
```

In this example, Student is the type and s1 is the reference variable that refers to the instance of Student class. The new keyword allocates memory at runtime.

C# Class

In C#, class is a group of similar objects. It is a template from which objects are created. It can have fields, methods, constructors etc.

Let's see an example of C# class that has two fields only.

```
public class Student
{
    int id;//field or data member
    String name;//field or data member
}
```

C# Object and Class Example

Let's see an example of class that has two fields: id and name. It creates instance of the class, initializes the object and prints the object value.

```
using System;

public class Student
{
    int id;//data member (also instance variable)
    String name;//data member(also instance variable)

    public static void Main(string[] args)
    {
        Student s1 = new Student();//creating an object of Student
        s1.id = 101;
        s1.name = "Sonoo Jaiswal";
        Console.WriteLine(s1.id);
        Console.WriteLine(s1.name);

    }
}
```

Output:

```
101
Sonoo Jaiswal
```

C# Class Example 2: Having Main() in another class

Let's see another example of class where we are having Main() method in another class. In such case, class must be public.

```
using System;

public class Student
{
    public int id;
    public String name;
}

class TestStudent{
    public static void Main(string[] args)
    {
        Student s1 = new Student();
        s1.id = 101;
        s1.name = "Sonoo Jaiswal";
        Console.WriteLine(s1.id);
        Console.WriteLine(s1.name);
    }
}
```

```
}  
}
```

Output:

```
101  
Sonoo Jaiswal
```

C# Class Example 3: Initialize and Display data through method

Let's see another example of C# class where we are initializing and displaying object through method.

```
using System;  
  
public class Student  
{  
    public int id;  
    public String name;  
    public void insert(int i, String n)  
    {  
        id = i;  
        name = n;  
    }  
    public void display()  
    {  
        Console.WriteLine(id + " " + name);  
    }  
}  
  
class TestStudent{  
    public static void Main(string[] args)  
    {  
        Student s1 = new Student();  
        Student s2 = new Student();  
        s1.insert(101, "Ajeet");  
        s2.insert(102, "Tom");  
        s1.display();  
        s2.display();  
    }  
}
```

Output:

101 Ajeet
102 Tom

C# Class Example 4: Store and Display Employee Information

```
using System;

public class Employee
{
    public int id;
    public String name;
    public float salary;
    public void insert(int i, String n, float s)
    {
        id = i;
        name = n;
        salary = s;
    }
    public void display()
    {
        Console.WriteLine(id + " " + name + " " + salary);
    }
}

class TestEmployee{
    public static void Main(string[] args)
    {
        Employee e1 = new Employee();
        Employee e2 = new Employee();
        e1.insert(101, "Sonoo", 890000f);
        e2.insert(102, "Mahesh", 490000f);
        e1.display();
        e2.display();
    }
}
```

Output:

101 Sonoo 890000
102 Mahesh 490000